Securing Mail Servers

Eduardo Takamura
ADNET Systems, Inc.
June 2003
(Rev. 8/2003)

Abstract

Almost every organization depends on electronic mail (email) systems to conduct business or to simply exchange information with people. Unfortunately, the same system that provides an efficient way to communicate with others can be exploited for malicious purposes if not configured properly. This concise document summarizes the basic items (in no particular order) that need to be taken into consideration by the system administrator to secure a general mail server. *Sendmail* on linux, a popular Mail Transport Agent (MTA), will be used to illustrate key points within this document.

SMTP Access Control I: Relaying Protection

The mail server administrator must know who is authorized to connect to the server and from where. Most users will be retrieving/sending messages remotely, that is, from a client computer as opposed to from the server itself.

Configure the mail server so that only authorized users may relay. In Sendmail, list the IP addresses of authorized relay machines in

```
/etc/mail/relay-domains
```

and restart the mail server daemon.

The MTA should deny any other third party relaying. Otherwise, if the mail server permits relaying, then the server is vulnerable to spam. (See *Fighting Spam I* and *Fighting Spam II* below)

SMTP Access Control II: Blocking Hostile Addresses

Some mail servers have the ability to deny incoming messages from specific email and/or IP addresses. This feature can be quite useful when the administrator needs to block hostile users (e.g., persistent spammers). To block an IP or email address using Sendmail's ACL, add the address to the file `/etc/mail/access` followed by the keyword "REJECT." For example:

```
johndoe@spammer.com      REJECT
123.456.678.901          REJECT
```

then issue the following command from the shell:

```
# makemap hash /etc/mail/access.db
< /etc/mail/access
```

Fighting Spam I: Third-Party Relay Protection

With the low cost of Internet access, more users are connecting to the Internet than ever before. This increasing Internet population is now becoming a target of ad bombardments via unsolicited commercial email (UCE). UCE, a.k.a. *spam*, is issued by certain merchants who see Internet users as potential consumers. These merchants not only send out unsolicited mass email, but also "steal" someone's mail server(s) to deliver those email messages for the sake of their anonymity.

Currently some states have passed anti-spam laws in the United States and in other countries, but they do not stop spammers from abusing open relay systems, and sending spam messages. One way the mail server administrator can deal with spammers is to ensure the mail server does not have open relays. In other words, no unauthorized user should be able to send/retrieve messages using the SMTP server. (See *SMTP Access Control I: Relaying Protection* above)

There are scripts that have been written to test if a mail server has open relays or not. They are available free of charge from several web sites. Also, there are sites that offer free third-party relay checks.

Fighting Spam II: Identifying Spam Messages

If the SMTP server is constantly receiving spam messages, the administrator might opt to install spam-detecting software to identify and re-route spam messages, and possibly block the spammer source address on the fly. There is a caveat though. Legitimate messages (i.e., non-spam messages) might be detected as spam message by the server.

One example of an anti-spam tool is SpamAssassin, an open-source application that uses a scoring system to identify spam messages. Messages classified as spam are tagged, and may be re-routed to reduce the amount of spam traffic sent to users.

Inoculating Mail Servers Against Viruses and Worms I: Filtering Attachments

There are several types of computer viruses in the wild today causing millions of damages to office and home computers worldwide. The ones that the mail server administrator is mostly concerned about though are those viruses/worms that propagate themselves through email.

While protecting client machines with anti-virus software is a good approach to safeguard the system and other computers against these threats, one cannot trust that users keep their anti-virus software updated with the latest virus definitions. Unless the system administrator ensures each individual client AV software is up-to-date, there is no guarantee that the client system and other machines on the network are protected.

On the other hand, the mail server administrator can protect all relaying clients by installing anti-virus software and/or implementing filters on the server.

The mail filters analyze mail headers in search for MIME attachments whose extensions match a list of prohibited extensions pre-defined by the administrator. One open-source tool to achieve this is MIMEDefang. This tool can also work with anti-spam tools, and it basically applies filters that inspect all incoming and outgoing messages, and quarantine attachments that match the filter rules. Once a message is quarantined, the sender, the receiver, and the system administrator are notified by email about the attachment file(s) that has/have been quarantined for inspection. The cleaned message is still delivered to the recipient(s).

There is a number of anti-virus software for mail servers commercially available on the market. Prices vary according to features and support. In terms of mail security, since most viruses/worms propagate themselves as a mail attachment, each having a known file extension (e.g., .pif, .scr, .vbs, .exe), it is more cost effective to implement filters on the SMTP server using the mentioned tool or other tools that accomplish the same job.

Inoculating Mail Servers Against Viruses and Worms II: Filtering Subject Lines

The better you know your enemy, the higher the chances of defeating him. Knowing about the virus (i.e., traces, payload, and other characteristics) and understanding how it works allow us to better equip ourselves against their threats. In the previous section, one method for fighting viruses is filtering messages containing attachments with specific file extensions. However, each time an infected message is processed by MIMEDefang, the sender, the recipient, and the mail administrator get a notice from the software. To avoid delivery of this "junk" mail, the mail server can be re-configured to simply reject those messages. One method to do so is to analyze the subject line in the mail header.

Viruses/worms often use the same subject lines. Security experts can quickly identify the various subject lines that are used by viruses/worms. [1] Knowing the contents of the subject line can be very useful in preventing infected messages from being delivered because we can re-configure the MTA to reject messages that match "prohibited" subject lines. For example, to reject all messages with subject "Re: Details" (W32.Sobig.F) and return an error 553 message back to the sender, add the following code to `/etc/mail/sendmail.mc`:

```
HSubject:
[TAB][TAB][TAB]$>Check_Subject
D{MPat}Re: Details
D{MMsg}THIS MESSAGE WAS NOT
DELIVERED TO THE RECIPIENT AS THE
SUBJECT LINE INDICATES PRESENCE OF
```

[1] You can find several resources on the Internet containing information about viruses and their characteristics. Some of those resources are listed at the end of this document.

```
VIRUS. PLEASE CHANGE THE SUBJECT
AND TRY AGAIN.
SCheck_Subject
R${MPat} $* [TAB] $#error $: 553
${MMsg}
RFW: ${MPat} $* [TAB] $#error $:
553 ${MMsg}
```

Use tabular indentation in place of [TAB]. Then issue the command:

```
m4 /etc/mail/sendmail.mc >
/etc/mail/sendmail.cf
```

and restart the SMTP daemon. Messages containing the same subject as you define will bounce back to the sender with an error 553 message.

The problem with sending error messages back to the sender, in this case, is the fact that some viruses spoof the sender and the reply-to addresses.

Review email aliases

Make sure email aliases point to authorized users/email addresses. Almost all MTAs are shipped with default or pre-defined email aliases. Customize the aliases file as appropriate. In Sendmail, the email alias file is located at `/etc/mail/aliases`. Syntax:

```
aliasname: username1, username3,
user@acme.com
```

Rebuild the aliases database by issuing the following command on the shell:

```
/usr/bin/newaliases
```

Encryption: Messages

Although not directly related to mail server security, it is worth mentioning that messages can be encrypted to protect the content of the data in transit. Public key encryption is one of the most commonly used methods for encrypting email messages. Resources available include *Pretty Good Privacy (PGP)* encryption, *Gnu Privacy Guard (GPG)*, etc.

Encryption: POP-3 Credentials

Client machines connect to the SMTP server though the Post Office Protocol (POP). When a POP connection is initiated by the user, the client sends its credentials (username and password) to the mail server to be authenticated prior to sending/ receiving messages. The credential information is transmitted in clear text to the server, hence the risk of a sniffer to pick up this sensitive information.
To protect the users, the system administrator may configure the SMTP server to use the encryption during the log in session. SSL/TLS use encryption algorithms for this protection.

Encryption: Webmail Access

If your mail server provides Webmail capabilities, ensure that at least the login page is encrypted with SSL. For further privacy protection of the user, enable SSL encryption through out the entire session. Caution: This might affect system performance.

Set Maximum Message Size

There is a slight possibility, depending on the configuration that the server might crash if it processes large mail messages, especially if the messages are sent to multiple recipients at once. To avoid this, set an appropriate maximum message size for your server. In Sendmail, edit the *MaxMessageSize* parameter in

```
/etc/mail/sendmail.cf
```

For example, to set the maximum message size to 15MB, set:

```
MaxMessageSize=15000000
```

Keep Software Up-to-Date

New bugs in software are found everyday, and one day your MTA software or mail client application may have a bug that could literally open a can of worms. So it is important to monitor new software releases and security bulletins, and to keep all software up-to-date.

Disable VRFY

The command VRFY can be used to verify if an email account exists on the server. The problem with allowing anyone to verify email addresses on the mail server is the fact that once an address has been verified, a username is automatically disclosed[*] to the verifier. Therefore, the server should not respond to VRFY requests. Disable VRFY (and EXPR) by

[*] Except in case an email alias is used.

editing the entry `PrivacyOptions` in `/etc/mail/sendmail.cf`:

```
PrivacyOptions=noexpn novrfy
```

Someone trying to gain access to a server may attempt to find out what user accounts are on the system. Knowing the username facilitates in 50% the chances of breaking into the system since all is left for the intruder is to launch a brute-force password attack on the guessed user account(s).

Monitor Mail Traffic Numbers

It is a good idea to monitor the traffic handled by the mail server to keep track of the number of messages processed by the mailer and to monitor the load of the mail server. By analyzing patterns in the mail load over time, the mail administrator may find clues about possible problems with the MTA. Example of such problems include unauthorized relaying and other misuse of the server.

Sendmail comes with a utility called *mailstats* that displays information about the mailers, the number and the size of transmitted/received messages, the number of rejected/discarded messages, etc.

Ensure that Sendmail is configured to log such statistical information by uncommenting the following in `/etc/mail/sendmail.mc`:

```
define(`STATUS_FILE',
`/etc/mail/statistics')
```

Then issue the following command:

```
m4 /etc/mail/sendmail.mc >
/etc/mail/sendmail.cf
```

Run `/usr/sbin/mailstats` at different time intervals to compare the load statistics.

Conclusion

Hardening a mail server is an on-going process that is worth investing. By securing an SMTP server, the administrator is indirectly safeguarding all client machines that connect to the server. One example of this "security inheritance" is the implementation of filtering mechanisms that detect and quarantine virus-infected attachment files at the mail server. The clients can neither receive nor send mail messages infected with viruses. No single security item discussed in this document is superior or more secure than another, but a combination of each of these items can provide several layers of security on the mail server.

Resources

CERT® Coordination Center
http://www.cert.org

European Institute for Computer Anti-Virus Research (EICAR) Anti-Virus test file
http://www.eicar.org/anti_virus_test_file.htm

EXPN & VRFY
http://www.burningvoid.com/iaq/expn-vrfy.html

Gnu Privacy Guard (GnuPG/GPG)
http://www.gnupg.org

McAfee Virus Information
http://us.mcafee.com/virusInfo/default.asp

Microsoft Antivirus Information
http://www.microsoft.com/security/antivirus

MIMEDefang:
http://www.mimedefang.com

Pretty Good Privacy (PGP)
http://www.pgpi.org/

SecurityFocus Virus Information
http://www.securityfocus.com/virus

Sendmail
http://www.sendmail.org

SpamAssassin
http://www.spamassassin.org

Symantec Virus Encyclopedia
http://securityresponse.symantec.com/avcenter/vinfodb.html

OpenSSL
http://www.openssl.org/

Third-Party Relay Check
http://www.rbl.jp/svcheck.php